

**REMARKS****Objection to the Abstract**

Applicants respectfully note that there is no express requirement that the abstract contain no numerical reference numbers. Applicants note that many issued US patents contain numerical reference numbers (see as examples, US6,240,528 B1; US5,829,412; and US6,748,558B1). As it is common for abstracts filed in the European patent office (EPO) to include numerical reference numbers, many people using both the EPO and the USPTO include numerical reference numbers in the abstract so that the same abstract may be used in both the EPO and USPTO.

**Objection to the Claims**

Although Applicants do not agree with the Examiner's objection to the use of "such as" in claim 10, Applicants have removed the clause containing these words from claim 10 in order to move prosecution of this case to resolution.

**Rejections under 35 U.S.C. 103**

Independent claims 1 and 14 were rejected under 35 U.S.C. 103(a) as being unpatentable over Leong (US publication 2003/0078902) in view of Garg (US patent 6,161,152), and further in view of Swenson (US patent 6,064,380).

**Regarding claim 1:**

Applicants respectfully note that Leong is solving a different problem in a different manner than the claimed invention. Leong teaches running JAVA on a device on top of another programming language, and accessing files through the interface between the two. Applicants respectfully note that Leong does not teach "a method of emulating more simultaneously open files in the device than is permitted by an operating system of the device". Leong does not address limitations on the number of open files; neither does Leong address emulating additional open files. Rather, the Leong patent teaches the wholesale conversion of data objects from one language into data objects for another language.

The Examiner argues that Leong teaches "while running a predetermined program in the JAVA programming language, attempting to access a file stored within the device". Applicants respectfully point out that the portions of Leong referenced by the Examiner deal with clients sending data to a server, whereas claim 1 deals with a single device. In claim 1, the JAVA program is running on the same device that the file is stored on. Leong does not teach this.

Applicants respectfully note that Garg is solving a different problem in a different manner than the claimed invention. The Examiner is correct that Garg teaches opening a file. Garg teaches opening a previously unopened file. Garg also teaches closing a file when a new file cannot be opened. However, Garg teaches opening a file based on an explicit request to do so (see Garg, FIG. 3, element 300). Thus, the Examiner is incorrect to state that Garg teaches the portions of claim 1 which read "if the file is open...." and "if the file is not open..." Nothing in Garg takes a read/write request and turns it into a file open operation if the file is not open.

Also, Garg does not teach emulating having more files open than the number allowed. Garg teaches the prior art idea of closing at least one of the open files to free a file descriptor (see Garg, col. 4, lines 21-25). Garg does not teach the emulation of having more files open than the number allowed by "(1) saving a file position for at least one open file, the file position designating where a next byte in the at least one open file would be accessed".

Applicants respectfully note that Swenson is solving a different problem in a different manner than the claimed invention. Swenson teaches user-initiated checkpointing of files. The "checkpointing" described in Swenson is a conscious action, in that the user is specifically asked if they want to checkpoint. As such, it is not "transparent" to the program. This is in direct conflict with claim 1 which states that "subsequent accesses to the ... open file that has been closed are made transparently to the ... program." In claim 1, the program neither requests to checkpoint the file nor requests to open it again. The program simply assumes that the file has been, and still is, open. Thus, as required by claim 1, subsequent accesses are made transparently to the program. Swenson does not teach this, and in fact teaches the opposite.

The Examiner asserts that it would have been obvious to combine Leong, Garg, and Swenson because all three references are devoted to frequent file data manipulation. Applicants respectfully point out that Swenson is not devoted to frequent file data manipulation in comparison with Leong and Garg. Leong and Garg deal with computer-computer interactions. Swenson teaches a relatively infrequent human-computer intervention that is very different from the frequent interactions that occur with a computer-computer interaction. Swenson teaches a person/user-driven process, and one could not expect the person/user to save the state of various files more than a couple times a minute (in practice, a person/user is more likely to do it only a couple times per day or week). Swenson's "checkpointing" process is a relatively infrequent human-computer intervention that is very different from the frequent interactions that occur with a computer-computer interaction. There is thus no motivation to combine these three references. These three references are also solving different problems in a different manner than the claimed invention and than each other, and thus there would be no motivation to combine all three of these references.

Regarding claim 14:

The same arguments from claim 1 apply. As previously discussed for claim 1, Garg only teaches opening a file when an express request to open a file is received, not generically when any I/O request (like read or write) comes in. In Garg, the file I/O request is either to open or to read/write. There is no notion in Garg, Leong, or Swenson of a request to just "read/write" resulting in opening the file.

The dependent claims are allowable for at least the same reasons as given for the independent claims.

Applicants believe the application is in condition for allowance which action is respectfully solicited. Please contact me if there are any issues regarding this communication or the current Application.

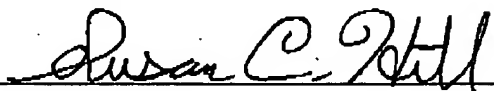
Respectfully submitted,

SEND CORRESPONDENCE TO:

Freescall Semiconductor, Inc.  
Law Department

Customer Number: 23125

By:



Susan C. Hill

Attorney of Record

Reg. No.: 35,896

Telephone: (512) 996-6839

Fax No.: (512) 996-6854

Email: Susan.Hill@Freescall.com